

Amendments to the Claims:

This listing of claims replaces all prior versions and listings of claims in the application:

Listing of Claims:

1. (Previously Presented) A method for embedding a foreign data block in a host data file, comprising:
 - receiving a host data file, the host data file having a host data file format;
 - receiving a foreign data block;
 - determining characteristics of the foreign data block, including a character encoding format of the foreign data block;
 - generating packing data that describes the characteristics of the foreign data block, including data identifying the beginning and end of the foreign data block and further including an identifier designed to be distinguishable from all other data in the host data file, wherein generating packing data includes selecting, based on the character encoding format of the foreign data block, a byte pattern that indicates a presence of a header, and including the byte pattern in the packing data; and
 - embedding the packing data and the foreign data block as a foreign data block packet in the host data file.
2. (Previously Presented) The method of claim 1, wherein:
 - generating packing data includes generating the header for the foreign data block, the header including the byte pattern and the identifier and indicating the beginning of the foreign data block packet and the beginning of the foreign data block.

3. (Original) The method of claim 2, wherein:
generating a header includes generating a header that indicates the end of the foreign data block packet.
4. (Original) The method of claim 2, wherein:
generating packing data includes generating a trailer for the foreign data block, the trailer indicating the end of the foreign data block.
5. (Original) The method of claim 2, wherein:
generating a header includes generating a header that indicates the end of the foreign data block.
6. (Original) The method of claim 1, further comprising:
including padding in the foreign data block packet to allow in place modifications of the foreign data block that cause the foreign data block to expand.
7. (Original) The method of claim 6, wherein:
determining characteristics of the foreign data block includes determining a size of the foreign data block; and
the amount of padding is a function of the size of the foreign data block.
8. (Original) The method of claim 1, wherein:
the foreign data block is a data block not native to the host file format.
9. (Original) The method of claim 1, wherein the foreign data block is an Extensible Markup Language (XML) document and the host file is in a non-XML format.

10. (Previously Presented) The method of claim 9, wherein:
determining characteristics of the foreign data block includes determining a byte order of the foreign data block; and

generating packing data includes generating information for specifying the byte order and character encoding format of the foreign data block, the character encoding format being one of an 8, 16, or 32 bit Unicode format.

11. (Original) The method of claim 1, wherein generating an identifier includes generating a different identifier for each different type of foreign data block when there are multiple types of foreign data blocks in the host data file.

12. (Original) The method of claim 1, wherein the foreign data block includes metadata information that describes the host data file.

13. (Original) The method of claim 12, wherein:
receiving a host data file includes receiving a host data file having a non XML format.

14. (Currently Amended) A computer program product, tangibly stored on a machine readable medium, comprising instructions operable to cause a programmable processor to:

receive a host data file; and

search for a header that indicates the beginning of an embedded foreign data block packet that contains a foreign data block, the foreign data block having a format that is recognizable by the computer program, the header including an identifier designed to be distinguishable from all other data in the host data file, the header further describing the characteristics of the foreign data block, wherein searching for the header comprises:

scanning byte by byte for a byte pattern that indicates a presence of a header; and

when the byte pattern is found, determining ~~determine~~ a character encoding format of the header and scanning character by character using the character encoding format to search for the identifier, and, if the identifier is found, processing the header or, if an identifier is not found, scanning a remaining portion of the host data file byte by byte for the byte pattern.

15. (Original) The computer program product of claim 14, further comprising instructions to:

process the foreign data block.

16. (Original) The computer program product of claim 15, further comprising instructions to:

stop processing the foreign data block when a trailer is detected, wherein the trailer indicates the end of the foreign data block.

17. (Original) The computer program product of claim 16, further comprising instructions to:

modify the foreign data block as specified by a user;

ensure that the modified foreign data block fits in the foreign data block packet; and

re embed the modified foreign data block in place of the original foreign data block.

18. (Original) The computer program product of claim 16, further comprising instructions to:

- modify the foreign data block as specified by a user;
- rewrite the foreign data block packet;
- ensure that the re written foreign data block packet is the same size as the original foreign data block packet; and
- re embed the re written foreign data block packet in place of the original foreign data block packet.

19-20. (Canceled)

21. (Previously Presented) A computer program product, tangibly stored on a machine readable medium, for embedding a foreign data block in a host data file, comprising instructions operable to cause a programmable processor to:

- receive a host data file, the host data file having a host data file format that is a native file format for a host application;

- receive a foreign data block, the foreign data block being a data block that is not native to the host data file format;

- determine characteristics of the foreign data block, including a character encoding format of the foreign data block;

- generate information that describes the characteristics of the foreign data block, including information identifying the beginning and end of the foreign data block and further including an identifier designed to be distinguishable from all other data in the host data file, wherein generating information identifying the beginning of the foreign data block includes selecting, based on the character encoding format, a byte pattern that indicates a presence of the information marking the beginning of the foreign data block; and

- embed the information and the foreign data block as a foreign data block packet in the host data file.

22. (Previously Presented) A computer program product, tangibly stored on a machine readable medium, for embedding metadata in a host data file having a non XML format, comprising instructions operable to cause a programmable processor to:

receive a host data file having a format that is not XML and that is a native file format for a host application;

receive metadata having a format that is not native to the host data file format;

determine characteristics of the metadata, including a character encoding format of the metadata;

generate information that describes the characteristics of the metadata, including information identifying the beginning and end of the metadata and further including an identifier designed to be distinguishable from all other data in the host data file, wherein generating information identifying the beginning of the metadata includes selecting, based on the character encoding format, a byte pattern that indicates a presence of the information marking the beginning of the metadata; and

embed the information and the metadata as a packet in the host data file.

23. (Previously Presented) The product of claim 21, further comprising instructions to:

generate a header for the foreign data block, the header including the identifier and indicating the beginning of the foreign data block packet and the beginning of the foreign data block.

24. (Previously Presented) The product of claim 21, further comprising instructions to:

generate a trailer for the foreign data block, the trailer indicating the end of the foreign data block.

25. (Previously Presented) The product of claim 21, further comprising instructions to:

include padding in the foreign data block packet to allow in place modifications of the foreign data block that cause the foreign data block to expand.

26. (Previously Presented) The product of claim 21, wherein the foreign data block is an Extensible Markup Language (XML) document and the host file is in a non-XML format.

27. (Previously Presented) The product of claim 26, further comprising instructions to:

determine a byte order of the foreign data block; and
generate a header that includes information for specifying the byte order and encoding format of the foreign data block, the encoding format being one of an 8, 16, or 32 bit Unicode format.

28. (Previously Presented) The product of claim 21, further comprising instructions to:

generate a different identifier for each different type of foreign data block when there are multiple types of foreign data blocks in the host data file.

29. (New) A method comprising:
receiving a host data file; and
searching for a header that indicates the beginning of an embedded foreign data block packet that contains a foreign data block, the foreign data block having a format that is recognizable by the computer program, the header including an identifier designed to be distinguishable from all other data in the host data file, the header further describing the characteristics of the foreign data block, wherein searching for the header comprises:
scanning byte by byte for a byte pattern that indicates a presence of a header; and
when the byte pattern is found, determining a character encoding format of the header and scanning character by character using the character encoding format to search for the identifier, and, if the identifier is found, processing the header or, if an identifier is not found, scanning a remaining portion of the host data file byte by byte for the byte pattern.

30. (New) The method of claim 29, further comprising:
processing the foreign data block.

31. (New) The method of claim 30, further comprising:
stop processing the foreign data block when a trailer is detected, wherein the trailer indicates the end of the foreign data block.

32. (New) The method of claim 31, further comprising:
modifying the foreign data block as specified by a user;
ensuring that the modified foreign data block fits in the foreign data block packet; and
re embedding the modified foreign data block in place of the original foreign data block.

33. (New) The method of claim 31, further comprising:
modifying the foreign data block as specified by a user;
rewriting the foreign data block packet;
ensuring that the re written foreign data block packet is the same size as the original
foreign data block packet; and
re embedding the re written foreign data block packet in place of the original foreign data
block packet.

34. (New) A method for embedding metadata in a host data file having a non XML
format, comprising:
receiving a host data file having a format that is not XML and that is a native file format
for a host application;
receiving metadata having a format that is not native to the host data file format;
determining characteristics of the metadata, including a character encoding format of the
metadata;
generating information that describes the characteristics of the metadata, including
information identifying the beginning and end of the metadata and further including an identifier
designed to be distinguishable from all other data in the host data file, wherein generating
information identifying the beginning of the metadata includes selecting, based on the character
encoding format, a byte pattern that indicates a presence of the information marking the
beginning of the metadata; and
embedding the information and the metadata as a packet in the host data file.

35. (New) The product of claim 23, further comprising instructions to:
generate a header that indicates the end of the foreign data block packet.

36. (New) The product of claim 23, further comprising instructions to:
generate a header that indicates the end of the foreign data block.

37. (New) The product of claim 25, wherein
determining characteristics of the foreign data block includes determining a size of the
foreign data block; and
the amount of padding is a function of the size of the foreign data block.

38. (New) The product of claim 21, wherein the foreign data block includes metadata
information that describes the host data file.

39. (New) The product of claim 38, further comprising instructions to:
receive a host data file having a non XML format.

40. (New) A system for embedding a foreign data block in a host data file,
comprising:

one or more computers operable to perform instructions to:

receive a host data file, the host data file having a host data file format;

receive a foreign data block;

determine characteristics of the foreign data block, including a character encoding
format of the foreign data block;

generate packing data that describes the characteristics of the foreign data block,
including data identifying the beginning and end of the foreign data block and further including
an identifier designed to be distinguishable from all other data in the host data file, wherein
generating packing data includes selecting, based on the character encoding format of the foreign
data block, a byte pattern that indicates a presence of a header, and including the byte pattern in
the packing data; and

embed the packing data and the foreign data block as a foreign data block packet
in the host data file.

41. (New) The system of claim 40, wherein:

generating packing data includes generating the header for the foreign data block, the header including the byte pattern and the identifier and indicating the beginning of the foreign data block packet and the beginning of the foreign data block.

42. (New) The system of claim 41, wherein:

generating a header includes generating a header that indicates the end of the foreign data block packet.

43. (New) The system of claim 41, wherein:

generating packing data includes generating a trailer for the foreign data block, the trailer indicating the end of the foreign data block.

44. (New) The system of claim 41, wherein:

generating a header includes generating a header that indicates the end of the foreign data block.

45. (New) The system of claim 40, wherein the one or more computers are further operable to perform instructions to:

include padding in the foreign data block packet to allow in place modifications of the foreign data block that cause the foreign data block to expand.

46. (New) The system of claim 45, wherein:

determining characteristics of the foreign data block includes determining a size of the foreign data block; and

the amount of padding is a function of the size of the foreign data block.

47. (New) The system of claim 40, wherein:

the foreign data block is a data block not native to the host file format.

48. (New) The system of claim 40, wherein the foreign data block is an Extensible Markup Language (XML) document and the host file is in a non-XML format.

49. (New) The system of claim 48, wherein:
determining characteristics of the foreign data block includes determining a byte order of the foreign data block; and
generating packing data includes generating information for specifying the byte order and character encoding format of the foreign data block, the character encoding format being one of an 8, 16, or 32 bit Unicode format.

50. (New) The system of claim 40, wherein generating an identifier includes generating a different identifier for each different type of foreign data block when there are multiple types of foreign data blocks in the host data file.

51. (New) The system of claim 40, wherein the foreign data block includes metadata information that describes the host data file.

52. (New) The system of claim 51, wherein:
receiving a host data file includes receiving a host data file having a non XML format.

53. (New) A system comprising:
one or more computers operable to perform instructions to:
receive a host data file; and
search for a header that indicates the beginning of an embedded foreign data block packet that contains a foreign data block, the foreign data block having a format that is recognizable by the computer program, the header including an identifier designed to be distinguishable from all other data in the host data file, the header further describing the characteristics of the foreign data block, wherein searching for the header comprises:
scanning byte by byte for a byte pattern that indicates a presence of a header; and
when the byte pattern is found, determining a character encoding format of the header and scanning character by character using the character encoding format to search for the identifier, and, if the identifier is found, processing the header or, if an identifier is not found, scanning a remaining portion of the host data file byte by byte for the byte pattern.

54. (New) The system of claim 53, wherein the one or more computers are further operable to perform instructions to:
process the foreign data block.

55. (New) The system of claim 54, wherein the one or more computers are further operable to perform instructions to:
stop processing the foreign data block when a trailer is detected, wherein the trailer indicates the end of the foreign data block.

56. (New) The system of claim 55, wherein the one or more computers are further operable to perform instructions to:
modify the foreign data block as specified by a user;
ensure that the modified foreign data block fits in the foreign data block packet; and
re embed the modified foreign data block in place of the original foreign data block.

57. (New) The system of claim 55, wherein the one or more computers are further operable to perform instructions to:

- modify the foreign data block as specified by a user;
- rewrite the foreign data block packet;
- ensure that the re written foreign data block packet is the same size as the original foreign data block packet; and
- re embed the re written foreign data block packet in place of the original foreign data block packet.

58. (New) A system for embedding metadata in a host data file having a non XML format, comprising:

- one or more computers operable to perform instructions to:
 - receive a host data file having a format that is not XML and that is a native file format for a host application;
 - receive metadata having a format that is not native to the host data file format;
 - determine characteristics of the metadata, including a character encoding format of the metadata;
 - generate information that describes the characteristics of the metadata, including information identifying the beginning and end of the metadata and further including an identifier designed to be distinguishable from all other data in the host data file, wherein generating information identifying the beginning of the metadata includes selecting, based on the character encoding format, a byte pattern that indicates a presence of the information marking the beginning of the metadata; and
 - embed the information and the metadata as a packet in the host data file.

59. (New) The method of claim 4, wherein the trailer further indicates whether the foreign data block packet is a read only packet.

60. (New) The method of claim 6, wherein the amount of padding is a function of an expected number of in place modifications of the foreign data block in the foreign data block packet or an expected number of foreign data block packets to be embedded in the host data file.

61. (New) The product of claim 24, wherein the trailer further indicates whether the foreign data block packet is a read only packet.

62. (New) The product of claim 25, wherein the amount of padding is a function of an expected number of in place modifications of the foreign data block in the foreign data block packet or an expected number of foreign data block packets to be embedded in the host data file.

63. (New) The system of claim 43, wherein the trailer further indicates whether the foreign data block packet is a read only packet.

64. (New) The system of claim 45, wherein the amount of padding is a function of an expected number of in place modifications of the foreign data block in the foreign data block packet or an expected number of foreign data block packets to be embedded in the host data file.